

Beyond Code: Elevating NextGen Statistical Programmers through Cross-Functional Learnings

Saumil Tripathi

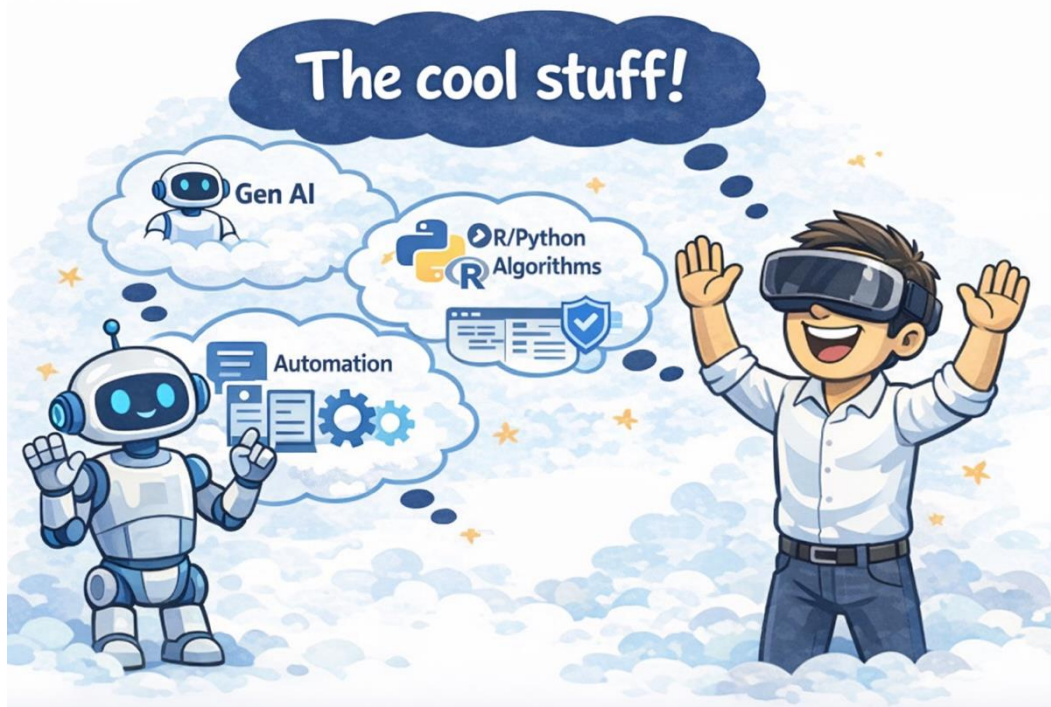
Associate Director – Biostatistics & Programming
Ephicity Lifescience Analytics, India.

Agenda

- Background
- Objective
- Clinical Research Workflow
- Case Study 1 - 3
- Cross Case Insights & Best Practices
- Conclusion
- Q & A

Background

What all Statistical Programmers EXPECT



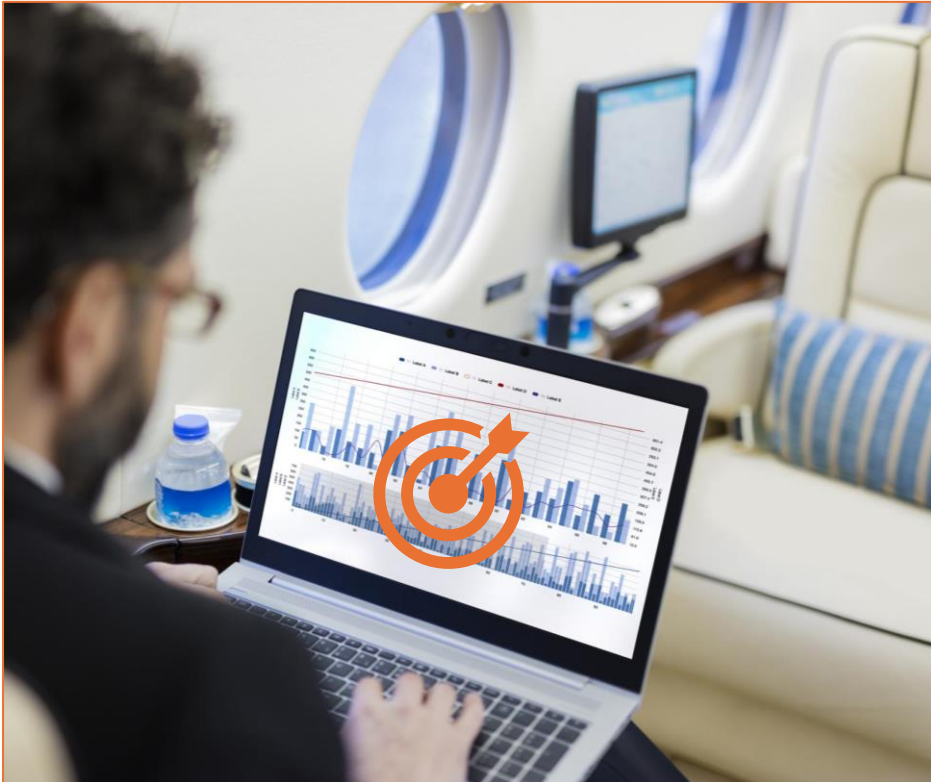
What is ACTUALLY Critical (Really ?)



Disclaimer : All skillsets are IMPORTANT !! Training priorities depends on context, timing and experience.

Background

- The role of statistical programmers is evolving beyond traditional coding into strategic clinical contributors.
- Increasing trial complexity driven by:
 - Multiple data sources (RWE, wearables, EHRs), Complex Study Designs
 - Stringent regulatory expectations (FDA/EMA)
- Limited understanding of cross-functional teams such as Clinical Operations, Data Management, and Regulatory Affairs leads to:
 - Communication gaps and delayed timelines
 - Data discrepancies, re-work, and protocol deviations
 - Regulatory queries due to non-compliant analysis datasets and outputs
- Cross-functional collaboration is critical to ensure data quality, compliance, and efficient drug development.



Elevating NextGen Statistical Programmer through Cross-Functional Learnings

Capability Expansion

- Statistical programmers gain new skills through cross-functional trainings in **Clinical Trial Operations**, **Data Management**, and **Regulatory Submissions**.

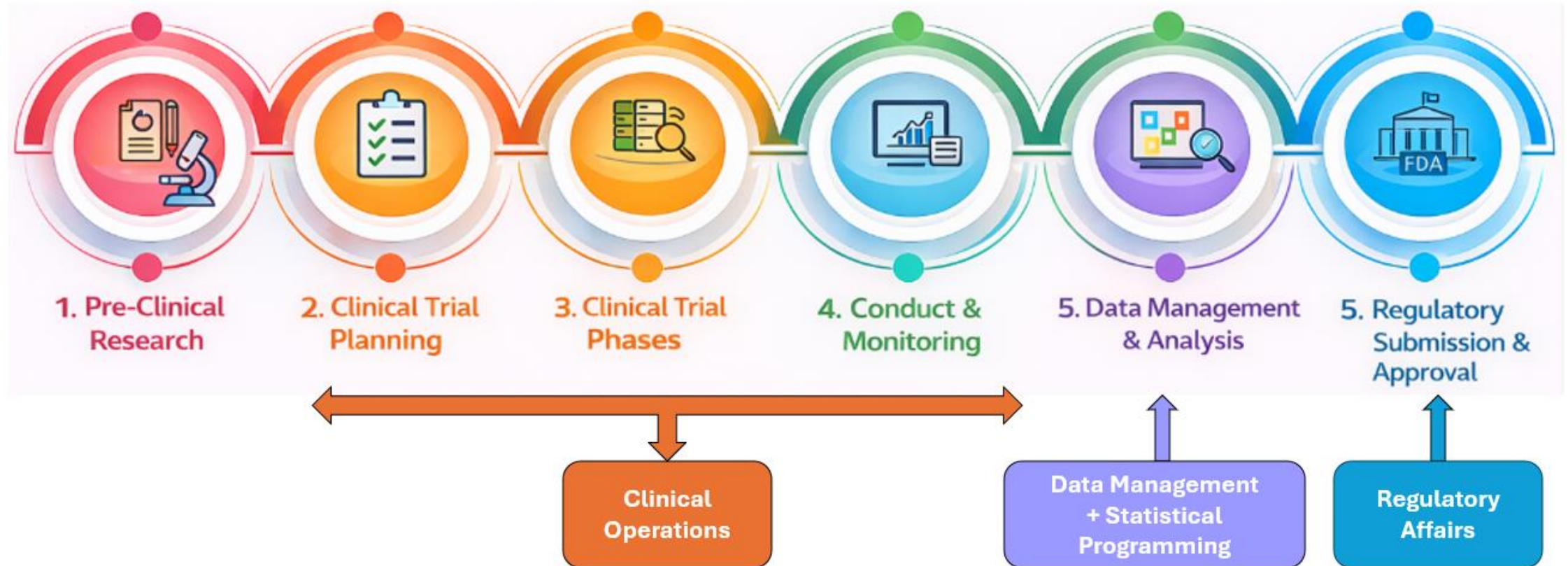
Value Creation

- Focus on anticipating stakeholder needs.
- Delivering submission-ready outputs.
- Reducing re-work and accelerating timelines for greater impact.

Long-term Impact

- Transformation enables career growth for programmers.
- Improves overall drug development efficiency.

Clinical Research Workflow



Case Study 1

Embedding Statistical Programmers in Clinical Operations: Turning Early Insight into Better Outcomes

Background

- Clinical operations decisions directly drive downstream data structure and analysis
- Late programming involvement increases risk of data issues, rework, and timeline delays
- Opportunity identified to embed programming team early to improve trial execution quality

Approach

- Early embedding of a lead statistical programmer with continuous cross-functional engagement
- Review of protocol and CRF using a data analysis lens
- Upskilling programmers in clinical operations fundamentals (study phases, data flow, GCP)

Outcome

- ✓ Prevented CRF design gaps before study start and Reduced rework during protocol amendments
- ✓ Mitigated analysis bias from early protocol deviation detection
- ✓ Improved data quality, compliance, and inspection readiness

Case Study 1

From Silos to Synergy: The Story of Embedded Clinical Programmers



Case Study 2

Shift Left to Move Faster: Enabling Faster Statistical Analysis Through Data Management Insights

Background

- Data management and programming teams are interdependent but functions separately
- Limited early visibility into upstream data capture and validation logic
- Misalignment often identified late, driving rework and delays
- Increased risk to data quality, compliance, and submission timelines



Approach

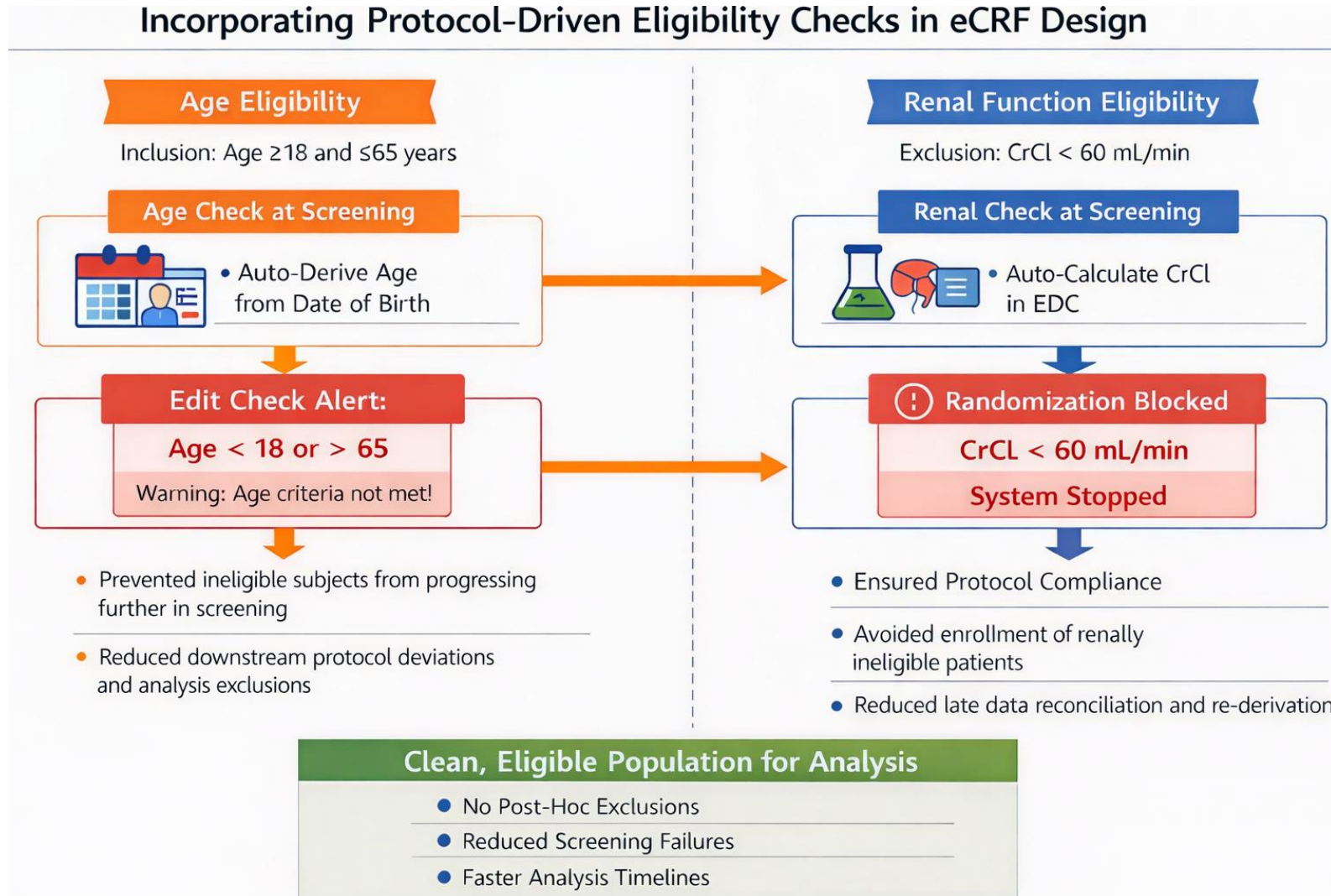
- Early review of eCRF and Data transfer guidelines by programming team
- Analysis-driven, real-time data queries review metrics
- Review of offline listings
- Protocol-driven eCRF controls to prevent protocol deviations
- Early lab and external data standardization for faster SDTM generation



Outcome

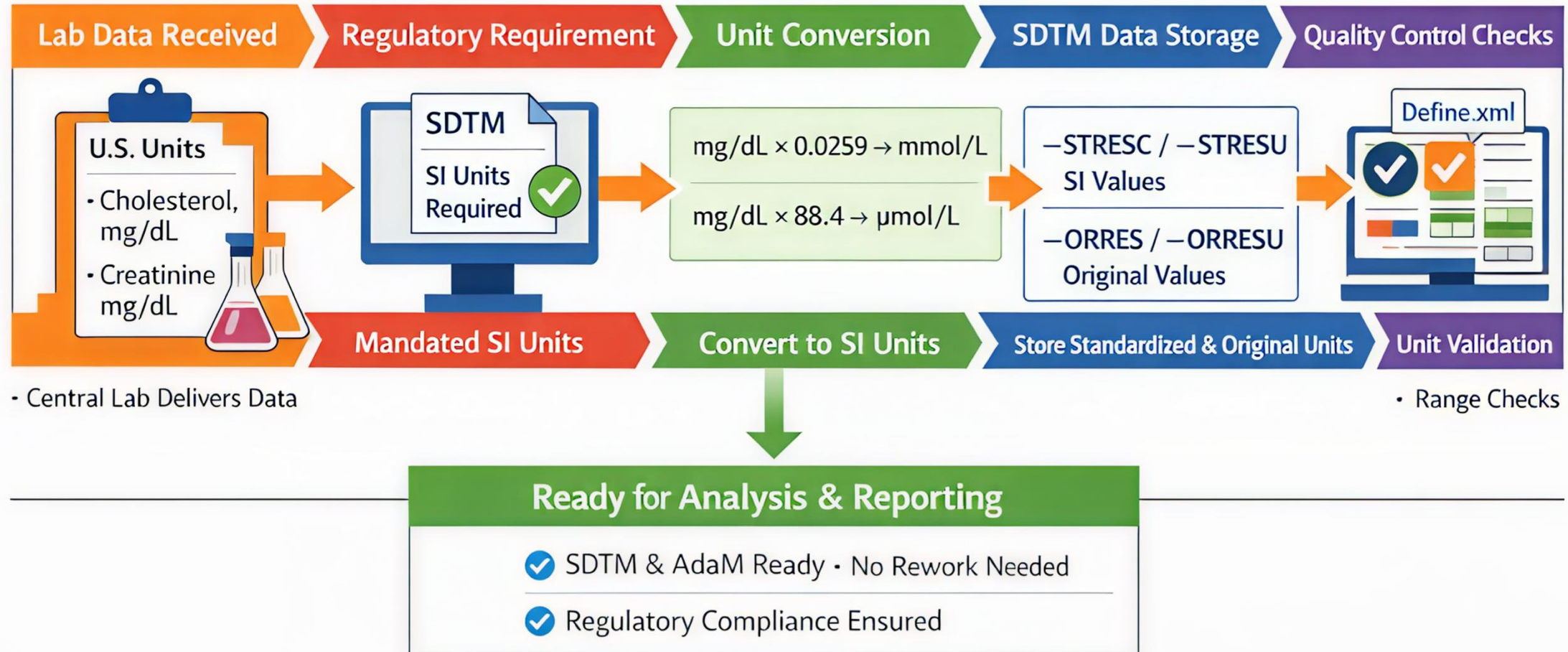
- ✓ Critical data issues identified and resolved earlier
- ✓ Decreased query turn around time with minimal pending queries at database lock
- ✓ Accelerated LPLV to Database lock timelines
- ✓ Minimal or no Re-work after DB Lock
- ✓ Improved overall data quality, efficiency, and submission readiness

Case Study 2



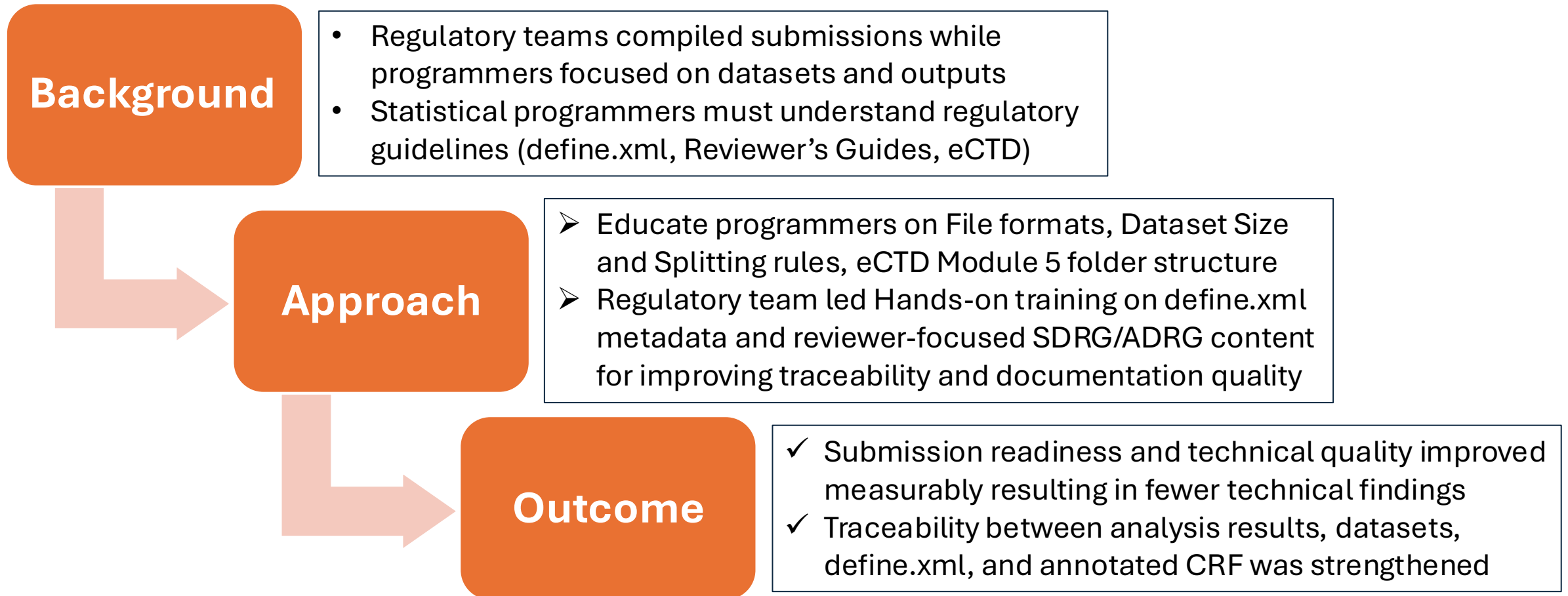
Case Study 2

Standardizing External Lab Data to SI Units for Statistical Analysis

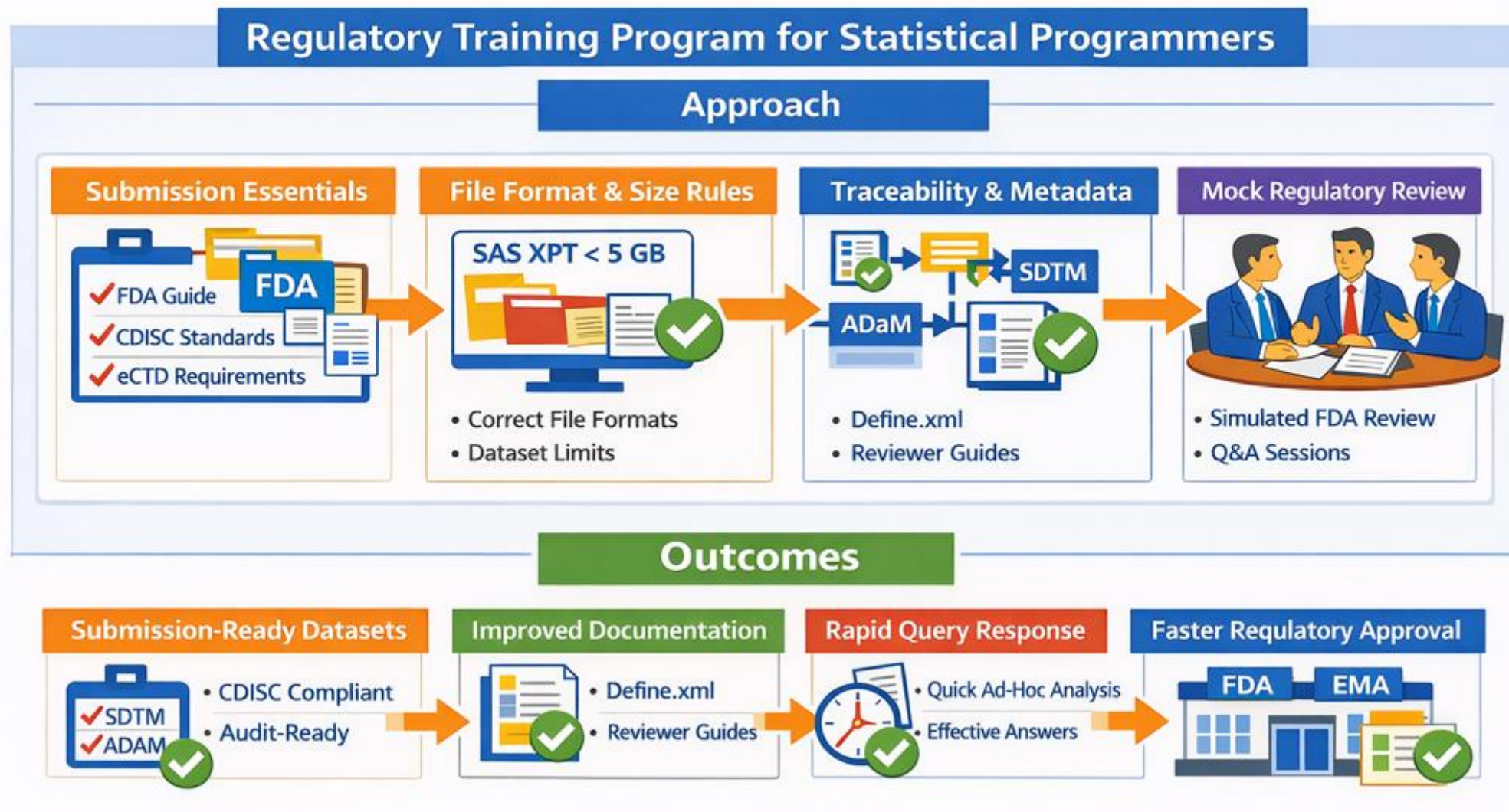


Case Study 3

Building a Submission-Ready Mindset: Cross-Functional Regulatory Training for Statistical Programmers



Case Study 3



Cross Case Insights & Best Practices

Insight Area	Impact on Statistical Programming	Best Practices
Early Cross-Functional Integration	• Reduced late-stage rework • Better stakeholder alignment	• Engage programmers early in protocol and CRF review, data strategy, and submission planning
Proactive Programming Mindset	• Accelerated programming timelines • Less re-work and high quality	• Position programmers as proactive contributors
Shared Data Quality Ownership	• Improved consistency, traceability, and interpretability	• Promote shared responsibility for data quality
Traceability & Compliance Focus	• Stronger and effective documentation • Smoother regulatory reviews	• Embed traceability principles into standards and training
Experiential Learning Models	• Sustained behavioral change • Improved programming efficiency	• Use rotations, shadowing, and collaborative reviews
Career Development Impact	• Greater strategic contribution to trials	• Formalize cross-functional learning in career progression

Conclusion

- ❖ Advanced technologies (GenAI, LLMs, Automation) are only effective when grounded in strong clinical and regulatory understanding, otherwise they risk amplifying compliance and data quality issues
- ❖ As regulatory expectations and trial complexity continue to increase, Cross-functional learning is a foundational requirement, not an enhancement for modern statistical programming
- ❖ Embedding exposure to clinical operations, data management, and regulatory affairs enables end-to-end lifecycle awareness, driving compliant, traceable, and submission-ready deliverables
- ❖ Elevating programmers beyond code transforms them into clinical data scientists, enabling strategic contributions to regulatory success, efficient trial execution, and data-driven decision making



Thank You !!

Any Questions

